Theoretical Computer Science 412 (2011) 1350–1363

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Digital distance functions on three-dimensional grids

Robin Strand [a,*], Benedek Nagy [b], Gunilla Borgefors [c]

[a] Centre for Image Analysis, Uppsala University, 75105 Uppsala, Sweden
[b] Department of Computer Science, Faculty of Informatics, University of Debrecen, Hungary
[c] Centre for Image Analysis, Swedish University of Agricultural Sciences, Sweden

ARTICLE INFO

Keywords:
Digital geometry
Distance functions
Distance transforms
Three-dimensional image processing
Non-standard grids

ABSTRACT

In this paper, we examine five different three-dimensional grids suited for image processing. Digital distance functions are defined on the cubic, face-centered cubic, body-centered cubic, honeycomb, and diamond grids. We give the parameters that minimize an error function that favors distance functions with low rotational dependency. We also give an algorithm for computing the distance transform—the tool by which these distance functions can be applied in image processing applications.

1. Introduction

Three-dimensional images generated by, e.g., tomographic methods are used as a diagnostic tool in medicine and increasingly in material science and other applications. With higher precision in image acquisition equipment, the amount of data that is used when processing these images is increasing. By using optimal sampling grids, almost 30% fewer samples are needed to get the same representation/reconstruction quality [1–3]. Alternatively, better quality can be obtained with the same number of samples. And as the images are computed, rather than captured directly, there is no special reason to use the standard cubic grid.

The possibility of using alternative structures for representing images in two, three, and higher dimensions has inspired researchers to develop image processing methods for non-standard grids [4–6]. One argument for using non-Cartesian grids is that fewer samples can be used to perfectly reconstruct a band-limited signal when sampled on some of these grids [1–3,7,8]. Given a grid for sampling, the density of the grid in the spatial domain is inversely proportional to the density in the frequency domain. The denser the grid in the frequency domain, the larger part of the frequency domain is covered by non-overlapping balls (which is required if aliasing should be avoided). Therefore, one important property when it comes to sampling functions efficiently is related to optimally dense packing. The packing density of a grid is defined as the fraction of the space that is covered by non-overlapping balls of maximal radius centered at the grid points. The densest three-dimensional grid is the face-centered cubic (fcc) grid [9]. The reciprocal grid (the grid in frequency domain) is the body-centered cubic (bcc) grid.

Acquiring images on non-standard grids by tomographic methods is considered in [10–12]. When reconstructing an object by tomography, each 2D slice of the 3D object that is being imaged is handled separately. If each such slice is reconstructed on a hexagonal grid and the slices are reconstructed without any lateral translation between the slices, the 3D volume image is represented on a honeycomb grid [8,13]. If lateral translation is applied, the 3D object can be represented by an fcc grid, since the hexagonal grid is embedded in the fcc grid [12].

An important and often used tool in image processing is the distance transform. In a distance transform, each background/object grid point is assigned the minimal distance from the object/background grid point. Some recent papers in which the distance transform is used in applications are found in [14–16].

* Corresponding author. Tel.: +46 18 4713469.
E-mail addresses: robin@cb.uu.se (R. Strand), nbenedek@inf.unideb.hu (B. Nagy), gunilla@cb.uu.se (G. Borgefors).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.10.027

Table 1
Packing densities of the grids considered here, in density order.

Grid	Packing density
fcc	$\pi/\sqrt{18} \approx 0.74$
bcc	$\pi\sqrt{3}/8 \approx 0.68$
Honeycomb	$\pi\sqrt{3}/9 \approx 0.60$
Cubic	$\pi/6 \approx 0.52$
Diamond	$\pi\sqrt{3}/16 \approx 0.34$

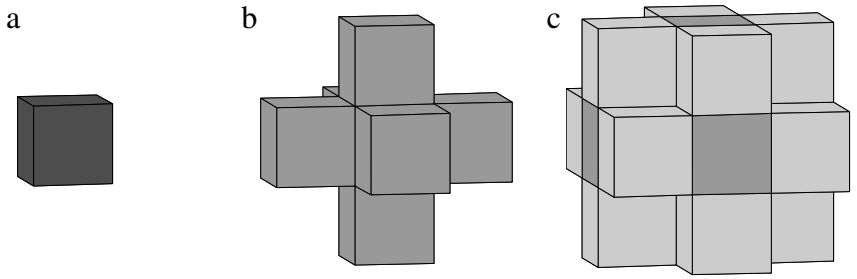

Fig. 1. (a) A voxel in the cubic grid, (b) voxels corresponding to 1-neighbors, and (c) 2-neighbors.

In this paper, we will consider the cubic, fcc, bcc, honeycomb, and diamond grids. All these grids, except the diamond grid, are *point-lattices*, meaning that the grid forms a discrete subgroup of Euclidean space and therefore can be described by a basis. It should also be noted that each of the cubic, fcc, bcc, and diamond grids is unique up to rotation and isotropic scaling. The honeycomb grid consists of layers that are hexagonal grids, and the height between each layer must be defined.

We will analyze distance functions defined as the minimal cost-path using two neighborhood relations on these five grids. With such path-based distance functions, the distance between two points is calculated by counting the number of (weighted) steps needed to go from one point to the other. We distinguish these *digital* distance functions from the Euclidean metric which is not discrete in this sense.

To define our digital distance functions, we use *weights* and *neighborhood sequences*. This framework includes weighted distances [17–20] and distances based on neighborhood sequences (ns-distances) [21–24], which are both generalizations of the well-known city block and chessboard metrics [25]. With weighted ns-distances [26–28], the rotational dependency of the distance function is potentially low [27,29].

In this paper, we build on some of our old results, for example [30,31] presented at the 13th International Workshop on Combinatorial Image Analysis (IWCIA 2009), and give formulas for weighted ns-distances for the five grids considered here. The performance on the different grids is compared by means of the asymptotic rotational dependency. The distance transform, the tool for using the distance functions in image processing, is described in Section 5.

2. Three-dimensional grids and neighborhood relations

The main motivation for this work is to allow image processing on non-standard, three-dimensional grids. In this section, we will define the five grids considered here and give some basic definitions on neighborhood relations. In the illustrations in this paper, each grid point is associated with a voxel defined as the Voronoi region of the grid at that point. The packing densities of these grids are found in Table 1.

2.1. The cubic grid

The cubic grid $\mathbb{Z}^3$ is often used for three-dimensional images. One reason is that there is a natural connection between the cubic grid and the data structure that is often used for storing images in computers.

Let (x, y, z) be the difference between the two points P and Q in $\mathbb{Z}^3$. The points P and Q are

$$\begin{aligned} &\text{1-neighbors if} && |x| + |y| + |z| = 1 \\ &\text{2-neighbors if} && \max(|x|, |y|, |z|) = 1 \text{ and } |x| + |y| + |z| = 2. \end{aligned}$$

See also Fig. 1.

Note that the 3-neighbor relations can also be defined in a natural way (i.e., based on Voronoi cells); ns-distances using all three types of neighborhood is described in [23,32–34].

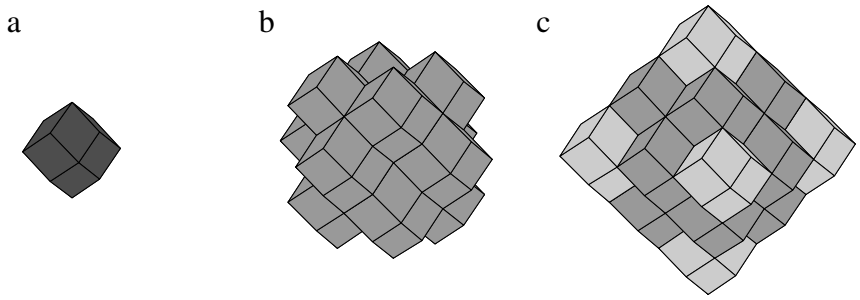

Fig. 2. (a) A voxel in the fcc grid, (b) voxels corresponding to 1-neighbors, and (c) 2-neighbors.

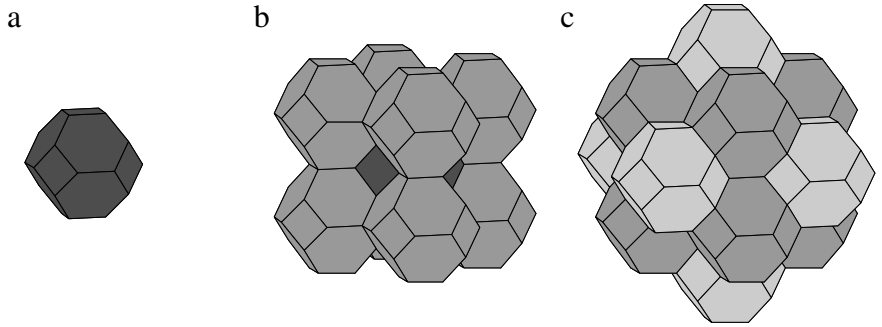

Fig. 3. (a) A voxel in the bcc grid, (b) voxels corresponding to 1-neighbors, and (c) 2-neighbors.

2.2. The face-centered cubic grid

The face-centered cubic (fcc) grid is the densest possible packing in three dimensions. This is the Kepler conjecture stated in 1611 [35], proved to be correct as late as 1998 by Hales [36]. Since the fcc grid is a densest packing, it has been considered for image processing in several papers. See for example [5,6,8]. In [37], it is proved that sampling a three-dimensional object (a binary function with smooth boundary) on an fcc grid gives better topology-preserving properties than sampling on a cubic grid with twice as many samples(!).

We use the following definition of the fcc grid: $\mathbb{F} = \{(x, y, z) \in \mathbb{Z}^3 \text{ and } x + y + z \equiv 0 \pmod 2\}$.

Let (x, y, z) be the difference between the two points P and Q in $\mathbb{F}$. The points P and Q are

$$\begin{aligned} \text{1-neighbors if} \quad & \max(|x|, |y|, |z|) = 1 \\ \text{strict 2-neighbors if} \quad & \max(|x|, |y|, |z|) = 2 \text{ and } |x| + |y| + |z| = 2. \end{aligned}$$

The shape of the voxels and the neighborhood relations are illustrated in Fig. 2.

2.3. The body-centered cubic grid

The body-centered cubic (bcc) grid is the reciprocal of the fcc grid. Therefore, it is optimal for sampling 3D functions according to the multidimensional Shannon sampling theorem [1–3,7,8]. It is defined as $\mathbb{B} = \{(x, y, z) \in \mathbb{Z}^3 \text{ and } x \equiv y \equiv z \pmod 2\}$.

Let (x, y, z) be the difference between the two points P and Q in $\mathbb{B}$. The points P and Q are

$$\begin{aligned} \text{1-neighbors if} \quad & \max(|x|, |y|, |z|) = 1 \\ \text{strict 2-neighbors if} \quad & \max(|x|, |y|, |z|) = 2 \text{ and } |x| + |y| + |z| = 2. \end{aligned}$$

As can be seen in Fig. 3, the voxels in the bcc grid are truncated octahedra and there are eight 1-neighbors and six strict 2-neighbors.

2.4. The honeycomb grid

The honeycomb grid $\mathbb{H}$ is obtained by piling hexagonal grids on top of each other. Therefore, it is natural to generate images on the honeycomb grid if each slice in a tomographic image acquisition method is represented by a hexagonal grid. In [13], digital geometry and computer graphics properties of the honeycomb grid are presented. The grid is spanned by the vectors $(1, 0, 0)$, $(0, 0, 1)$, $\left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right)$.

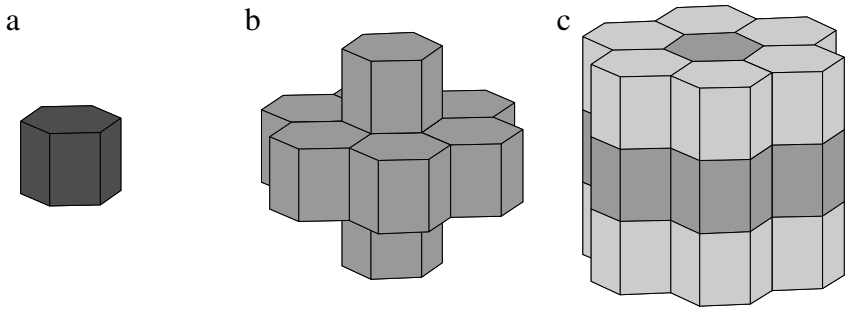

Fig. 4. (a) A voxel in the honeycomb grid, (b) voxels corresponding to 1-neighbors, and (c) 2-neighbors.

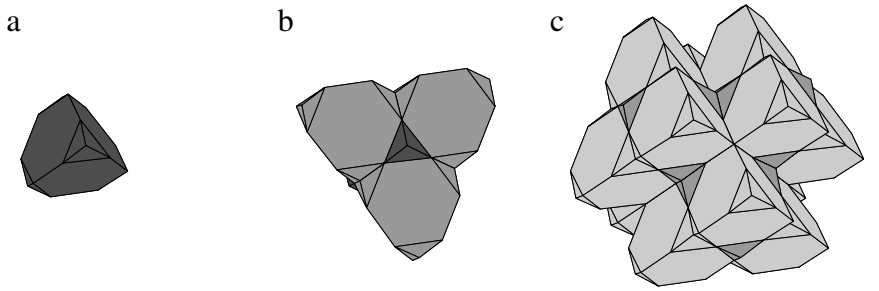

Fig. 5. (a) A voxel in the diamond grid, (b) voxels corresponding to 1-neighbors, and (c) 2-neighbors.

Let (x, y, z) be the difference between the two points P and Q in $\mathbb{D}$. The points P and Q are

$$\begin{aligned} \text{1-neighbors if} \quad & |(x, y, z)| = 1 \\ \text{strict 2-neighbors if} \quad & |(x, y, z)| = \sqrt{2}. \end{aligned}$$

The neighborhoods are illustrated in Fig. 4. Note that $(1, 0, 0)$, $(0, 0, m)$, $\left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right)$ generates a honeycomb grid for any m, where m is the distance between the hexagonal layers. We use $m = 1$ here since this has been shown to be optimal for digital (weighted) distance functions in [12].

2.5. The diamond grid

The diamond grid, or tetrahedral packing, is found in diamond which is the hardest substance known. It is not a point-lattice, but there is a natural relation between the cubic, fcc, bcc, and the diamond grids [38,39]. You can say that in the same way that the cubic grid is the 3D grid that corresponds to the 2D square grid and the fcc and bcc grids correspond to the 2D hexagonal grid, the diamond grid corresponds to the 2D triangular grid.

A definition of the diamond grid is $\mathbb{D} = \{(x, y, z) \in \mathbb{Z}^3, x \equiv y \equiv z \pmod 2 \text{ and } x + y + z \in \{0, 1\} \pmod 4\}$. Which neighboring points a point in the diamond grid has depends on its *parity*: a point in the diamond grid is even if $x+y+z \equiv 0 \pmod 4$. Otherwise it is an odd point. Let (x, y, z) be the difference between the two points P and Q in $\mathbb{D}$. The points P and Q are

$$\begin{aligned} \text{1-neighbors if} \quad & \max(|x|, |y|, |z|) = 1 \\ \text{strict 2-neighbors if} \quad & \max(|x|, |y|, |z|) = 2 \text{ and } |x| + |y| + |z| = 4. \end{aligned}$$

In Fig. 5, the neighborhood relations are illustrated.

We note here that the 1-neighborhood plays the role of chemical bonds in diamond (and other materials using this crystal structure). The strict 2-neighborhood has also chemical meaning, since this relation represents the closest same type atoms in GaAs and in some other materials. The 3-neighbor relations can also be defined using Voronoi cells [40]. Moreover, the 4-neighborhood can be defined in natural way by a unit cell [38,39,31]. However in this paper we generalize the ns-distances to wns-distances only for the first two neighborhood.

3. Distance functions

The distance functions we consider here are defined as minimal cost-paths. This is a property which makes them well-suited for efficient digital image processing algorithms, as we will see in Section 5. Since the weighted ns-distances are

generalizations of the weighted distances and ns-distances, the formulas presented here also gives the weighted distance and the ns-distance.

Two points are *adjacent* if they are 2-neighbors. A neighborhood sequence (ns) B is a sequence $B = (b(i))_{i=1}^{\infty}$, where each $b(i)$ denotes a neighborhood relation in a grid $\mathbb{G}$. If B is periodic with period l, then we write $B = (b(1), b(2), \ldots, b(l))$. A *path* in a grid is a sequence of adjacent grid points. A path $P_0, P_1, \ldots, P_n$ is a *B-path* of length n if, for all $i \in \{1, 2, \ldots, n\}$, P_{i-1} and P_i are $b(i)$-neighbors.

Definition 1. Given the ns B, the ns-distance $d(P_0, P_n; B)$ between the P_0 and P_n is the length of (one of) the shortest B-path(s) between the points.

Let the real numbers α and β (the *weights*) and a path of length n, where exactly l ($l \leq n$) adjacent grid points in the path are strict 2-neighbors, be given. The *cost* of the path is $(n-l)\alpha + l\beta$. A B-path between P_0 and P_n is a *minimal cost* B-path if no other B-path between the points has lower cost.

Definition 2. Given the ns B and the weights α, β, the weighted ns-distance $d_{\alpha,\beta}(P_0, P_n; B)$ is the cost of a minimal cost B-path(s) between the points.

The following notation is used:

$$\mathbf{1}_B^k = |\{i : b(i) = 1, 1 \leq i \leq k\}| \quad \text{and} \quad \mathbf{2}_B^k = |\{i : b(i) = 2, 1 \leq i \leq k\}|.$$

In the formulas below, the weights α, β are such that $\alpha \leq \beta \leq 2\alpha$. We remark that the weighted distance is obtained when $B = (2)$ and the ns-distance is obtained when $\alpha = \beta = 1$. The formulas are valid for any two points P, Q in the grid such that $x \geq y \geq z \geq 0$, where $(x, y, z) = Q - P$.

3.1. The cubic grid

Weighted ns-distances in $\mathbb{Z}^3$ were examined in [28]. In [8,34,41,33], formulas for ns-distances (with unit weights) were presented for the cubic grid $\mathbb{Z}^3$. We will use these results to give a formula for weighted ns-distances in $\mathbb{Z}^3$.

The following formula is a slightly modified version of Corollary 3.5 in [8].

Lemma 1 (*ns-distance in $\mathbb{Z}^3$*)**.** *The ns-distance is given by*

$$d(P, Q; B) = \begin{cases} \min\limits_k \left\{k \geq \max\left\{x+y+z-\mathbf{2}_B^k\right\}\right\} & \text{if } x < y+z \text{ and } (x,y,z) \in \mathbb{F} \\ \min\limits_k \left\{k \geq 1+\max\left\{x+y+z-1-\mathbf{2}_B^k\right\}\right\} & \text{if } x < y+z \text{ and } (x,y,z) \notin \mathbb{F} \\ \min\limits_k \left\{k \geq \max\left\{x, x+y+z-\mathbf{2}_B^k\right\}\right\} & \text{if } x \geq y+z. \end{cases}$$

Theorem 1 (*Weighted ns-distance in $\mathbb{Z}^3$*)**.** *The weighted ns-distance is given by*

$$d_{\alpha,\beta}(P, Q; B) = (2d(P, Q; B) - d(P, Q; (1)))\,\alpha + (d(P, Q; (1)) - d(P, Q; B))\,\beta.$$

Proof. When $x, y, z \geq 0$, $(x, y, x) = a_1(1, 0, 0) + a_2(0, 1, 0) + a_3(0, 0, 1)$ for some a_1, a_2, a_3. The 1-distance is $a_1 + a_2 + a_3$. If the ns B allows b 2-steps to reach (x, y, z), then there are $a_1', a_2', a_3', b_1, b_2, b_3$ such that the B-distance is given by $d = a_1' + a_2' + a_3' + b_1 + b_2 + b_3$, where $(x, y, z) = a_1'(1, 0, 0) + a_2'(0, 1, 0) + a_3'(0, 0, 1) + b_1(1, 1, 0) + b_2(0, 1, 1) + b_3(1, 0, 1)$, where all coefficients are non-negative and $b \geq b_1 + b_2 + b_3$. We now prove that this corresponds to a shortest B-path with minimal number of 2-steps:
Assume that there is a shortest B-path with smaller number of 2-steps, then some additional local steps are used in such a path and since all other local steps have at least one negative coordinate value, either the number of steps is larger than d or the number of steps equals d and the number of 2-steps is larger than $b_1 + b_2 + b_3$, which means that the assumption is false.

Since we have a shortest B-path with minimal number of 2-steps, for any weights α, β such that $\alpha \leq \beta \leq 2\alpha$, the weighted ns-distance is $(\sum a_i')\alpha + (\sum b_i)\beta$. The formula follows from the identity $d(P, Q; (1)) = \sum a_i = \sum a_i' + 2\sum b_i = d(P, Q; B) + \sum b_i$, where $\sum b_i$ is the number of 2-steps in the shortest B-path. □

3.2. The face- and body-centered cubic grids

The following theorems were first presented as Theorems 1 and 2, respectively, in [27]. They are also found in [8,42].

Theorem 2 (*Weighted ns-distance in $\mathbb{F}$*)**.** *The weighted ns-distance is given by*

$$d_{\alpha,\beta}(P, Q; B) = \begin{cases} k \cdot \alpha & \text{if } x \leq y+z \\ (2k-x)\cdot\alpha + (x-k)\cdot\beta & \text{otherwise,} \end{cases}$$

$$\text{where } k = \min_l : \left\{ l \geq \max\left(\frac{x+y+z}{2}, x - \mathbf{2}_B^l\right)\right\}.$$

Theorem 3 (*Weighted ns-distance in* $\mathbb{B}$). *The weighted ns-distance is given by*

$$d_{\alpha,\beta}(P, Q; B) = (2k - x) \cdot \alpha + (x - k) \cdot \beta$$
$$\text{where } k = \min_l : \left\{ l \geq \max\left(\frac{x+y}{2}, x - \mathbf{2}_B^l \right) \right\}.$$

3.3. The honeycomb grid

The following theorem gives the weighted ns-distance between points in the honeycomb grid. It is a slightly modified version of Corollary 3.14 in [8]. To make the formula valid for $x \geq y \geq z \geq 0$, we note that when $\frac{y}{x} > \frac{1}{\sqrt{3}}$ the distance value is obtained by changing the coordinates using a rotation by $\pi/3$ in the xy-plane.

Theorem 4 (*Weighted ns-distance in* $\mathbb{H}$). *The weighted ns-distance is given by*

$$d_{\alpha,\beta}(P, Q; B) = (2k - T - z) \cdot \alpha + (T + z - k) \cdot \beta$$

where

$$k = \min_l : l \geq \max\left(T, z, T + z - \mathbf{2}_B^l\right)$$

and

$$T = \begin{cases} x + \frac{1}{\sqrt{3}} y & \text{if } 0 \leq \frac{y}{x} \leq \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{3}} y & \text{otherwise.} \end{cases}$$

3.4. The diamond grid

Distances based on neighborhood sequences (i.e., without weights) are examined in [40,31,43]. Since the symmetries are not as obvious as on the other grids, we give formulas that hold for arbitrary P, Q. We still use the notation $Q - P = (x, y, z)$. The following lemma is presented as Theorem 1 in [40].

Lemma 2 (*ns-distance in* $\mathbb{D}$). *If P and Q have the same parity , then*

$$d(P, Q; B) = \min_l : \sum_{i=1}^{l} b(i) \geq \left\{ \frac{|x| + |y| + |z|}{2}, |x|, |y|, |z| \right\}.$$

If

- *P is even, Q is odd and (x, y, z) has an odd number of negative values or*
- *P is odd, Q is even and (x, y, z) has an even number of negative values*

then

$$d(P, Q; B) = \min_l : \sum_{i=1}^{l} b(i) \geq \left\{ \frac{|x| + |y| + |z| - 1}{2}, |x|, |y|, |z| \right\}.$$

If

- *P is even, Q is odd and (x, y, z) has an even number of negative values or*
- *P is odd, Q is even and (x, y, z) has an odd number of negative values*

then

$$d(P, Q; B) = \min_l : \sum_{i=1}^{l} b(i) \geq \left\{ \frac{|x| + |y| + |z| + 1}{2}, |x|, |y|, |z| \right\}.$$

Remark 1. The diamond grid has the following property: two points $P \neq Q$ are 2-neighbors iff there is a 1-path of length two between the points.

Theorem 5 (*Weighted ns-distance in* $\mathbb{D}$). *The weighted ns-distance is given by*

$$d_{\alpha,\beta}(P, Q; B) = (2d(P, Q; B) - d(P, Q; (1)))\,\alpha + (d(P, Q; (1)) - d(P, Q; B))\,\beta.$$

Proof. We prove that, any shortest B-path between two points P and Q is also a minimal cost-path with weights α, β such that $0 \leq \alpha \leq \beta \leq 2\alpha$:
If it is not a minimal cost-path, then there is a path with lower cost (but not shorter) than the given path. This means that the number of strict 2-steps is larger, but by Remark 1, any B-path consists if as many strict 2-steps as possible. Therefore, the

ns-distance given in Lemma 2 can be used to define also the weighted ns-distance. The formula follows from the identity given in the proof of Theorem 1. □

4. Parameter optimization

Finding optimal parameters for digital distance functions has been considered by many authors using different error functions [8,18–20,26,44–48]. The goal is to find parameters that give minimal rotational dependency for the digital distance functions. In this way, we can approximate the Euclidean distance without leaving the framework of digital distance functions.

In this section, we use the compactness measure

$$E = \frac{A^3}{V^2}\frac{1}{36\pi} - 1,$$

where V is the volume and A is the area of the boundary of a polyhedron. The polyhedra used here correspond to the asymptotic shapes of the digital balls generated by the distance functions presented in Section 3.

In [49], we showed that the compactness measure is close to the asymptotic optimal also for short distances on the fcc and bcc grids. Experiments have shown that this is the case also for the other grids considered here. However, we focus on the asymptotic behavior in this paper.

The asymptotic compactness measure of balls generated by the distance functions are obtained by using continuous versions of the discrete quantity $\mathbf{1}_B^k$ and allowing $\mathbb{R}^3$ as the domain for the distance functions. We replace $\lim_{k\to\infty} \mathbf{1}_B^k/k$ by a parameter, γ. Note that $\lim_{k\to\infty} \mathbf{2}_B^k/k$ corresponds to $(1-\gamma)$. The polyhedra corresponding to the asymptotic shape of the digital balls are obtained in this way. The shapes of these polyhedra are used in the optimization.

4.1. The cubic grid

The formulas in Lemma 1 and Theorem 1 are rewritten using $\gamma t,\ 0 \le \gamma \le 1, t \in \mathbb{R}$ instead of $\mathbf{1}_B^k$. The continuous value t is used instead of the discrete value k to have a distance functions defined for any $(x, y, z) \in \mathbb{R}^3$ such that $x \ge y \ge z \ge 0$. We get

$$d = (2t - x - y - z)\,\alpha + (x + y + z - t)\,\beta, \text{ where}$$
$$t = \begin{cases} x + y + z - (1-\gamma)t & \text{if } x < y + z \\ \max\{x, x + y + z - (1-\gamma)t\} & \text{if } x \ge y + z. \end{cases}$$

To get the surface patches that bound the polyhedron, we will solve $d = r$, where r is some radius.

4.1.1. Case i: $x < y + z$

Now $t = x + y + z - (1-\gamma)t$, i.e., $t = \frac{x+y+z}{2-\gamma}$ which gives $r = \frac{x+y+z}{2-\gamma}(2\alpha - \beta) + (x + y + z)(\beta - \alpha)$.

4.1.2. Case ii: $x \ge y + z$ and $x \ge x + y + z - (1-\gamma)t$

$t = x$, so $r = (x - y - z)\alpha + (y + z)\beta$.

4.1.3. Case iii: $x \ge y + z$ and $x \le x + y + z - (1-\gamma)t$

Now $t = x + y + z - (1-\gamma)t$, i.e., $t = \frac{x+y+z}{2-\gamma}$ which gives $r = \frac{x+y+z}{2-\gamma}(2\alpha - \beta) + (x + y + z)(\beta - \alpha)$.

We note that the shape of the polyhedron depends on β/α and γ. The formulas above describe planes which bound a polyhedron in the region $x \ge y \ge z \ge 0$. By symmetry, the entire polyhedron is described. The vertices are, up to symmetry, given by

$$\frac{r}{a}(1, 0, 0) \quad \text{and} \quad \frac{r}{\beta + \alpha\gamma - \beta\gamma}(1, 1-\gamma, 0).$$

The optimal values $\frac{\beta}{\alpha} \approx 1.307$ and $\gamma = 0$ are obtained by optimizing E with respect to these two parameters. These values give $E \approx 0.137$. Polyhedra for some parameters are shown in Fig. 6. The polyhedron for the parameters that minimize E is shown in Fig. 11.

4.2. The fcc and bcc grids

The optimization for the fcc and bcc grids are found in [8,27,29]. The shape of polyhedra for some parameters are found in Figs. 7 and 8. Optimal values are found in Table 2. The shape of the "optimal" polyhedra are shown in Fig. 11.

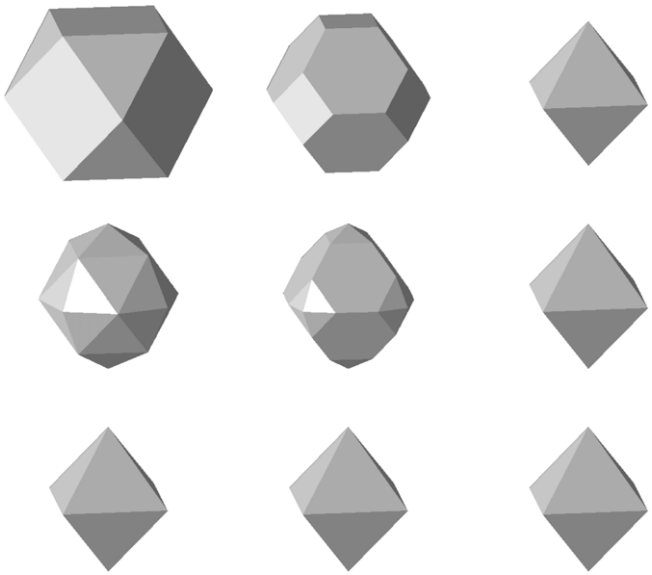

Fig. 6. Asymptotic shapes of balls in the cubic grid. Left to right: $\gamma = 0, 0.5, 1$. Top to bottom: $\beta/\alpha = 1, 1.5, 2$.

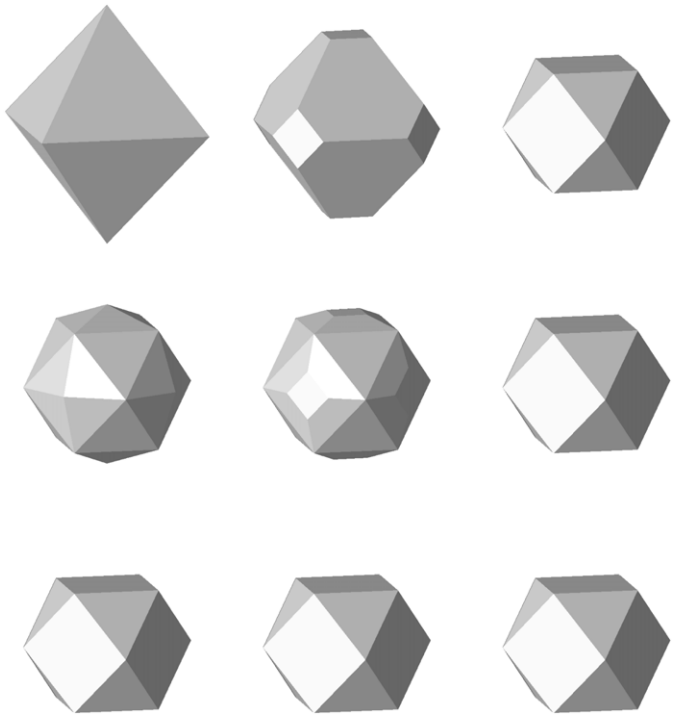

Fig. 7. Asymptotic shapes of balls in the fcc grid. Left to right: $\gamma = 0, 0.5, 1$. Top to bottom: $\beta/\alpha = 1, 1.5, 2$.

4.3. The honeycomb grid

By replacing $\mathbf{1}_B^k$ by γt in Theorem 4 and rewriting the formula, we get

$$
\begin{aligned}
d &= (2t - T - z) \cdot \alpha + (T + z - t) \cdot \beta, \text{ where} \\
t &= \max(T, z, T + z - (1 - \gamma)t) \text{ and} \\
T &= \begin{cases} x + \frac{1}{\sqrt{3}} y & \text{if } 0 \le \frac{y}{x} \le \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{3}} y & \text{otherwise.} \end{cases}
\end{aligned}
$$

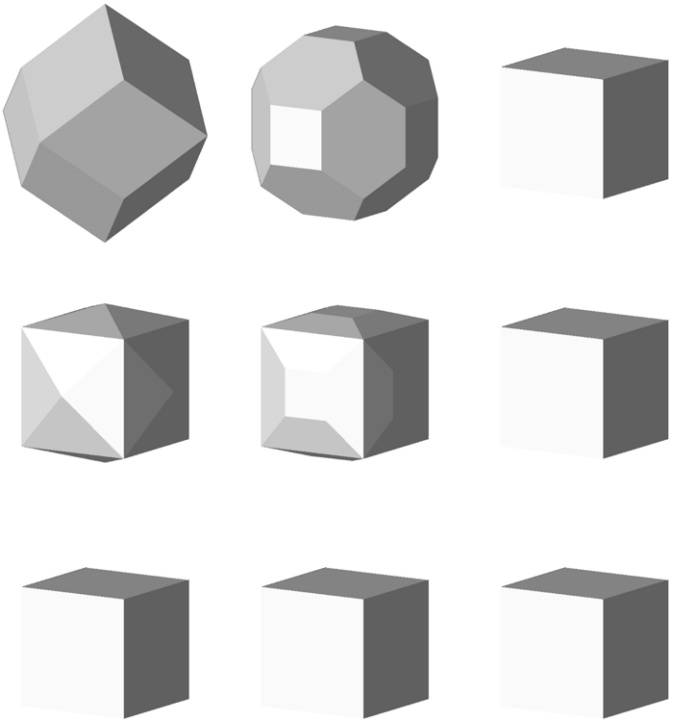

Fig. 8. Asymptotic shapes of balls in the bcc grid. Left to right: $\gamma = 0, 0.5, 1$. Top to bottom: $\beta/\alpha = 1, 1.5, 2$.

4.3.1. Case i: $T \geq z, T + z - (1-\gamma)t$
$r = (T - z)\alpha + z\beta$.

4.3.2. Case ii: $z \geq T, T + z - (1-\gamma)t$
$r = (z - T)\alpha + T\beta$.

4.3.3. Case iii: $T + z - (1-\gamma)t \geq T, z$
In this case $t = \frac{T+z}{2-\gamma}$, which gives $r = \left(\left(\frac{2}{2-\gamma} - 1\right)\alpha + \left(1 - \frac{1}{2-\gamma}\right)\beta\right)(T + z)$.

The shapes of the polyhedra bounded by the planes corresponding to the formulas calculated for the three cases are shown in Fig. 9 for some parameters. The values that minimize E are found in Table 2 and for the corresponding polyhedron, see Fig. 11.

4.4. The diamond grid

We use the identity $\sum_{i=1}^{k} b(i) = k + \mathbf{2}_B$ before γt is used to replace $\mathbf{1}_B^k$ in Lemma 2. Asymptotically, the parity of the points does not matter, so we use the formula for points of the same parity and get

$$\begin{aligned} d &= (2t - t')\alpha + (t' - t)\beta, \text{ where} \\ t : t + (1-\gamma)t &= \max\left\{\frac{|x| + |y| + |z|}{2}, |x|, |y|, |z|\right\} \text{ and} \\ t' &= \max\left\{\frac{|x| + |y| + |z|}{2}, |x|, |y|, |z|\right\}. \end{aligned}$$

This leads to

$$t = \frac{t'}{2-\gamma}, \text{ so } d = \left(\frac{\gamma}{2-\gamma}\alpha + \frac{2}{2-\gamma}\beta\right)t'.$$

Adjusting the parameters results in a scaling of the polyhedron, so any parameters give the optimal value $E \approx 0.349$. By Remark 1, any neighborhood sequence and weights result in the same polyhedron—a cuboctahedron, see Figs. 10 and 11. This interesting phenomenon can also be connected to the fact that the diamond grid is not a lattice, and therefore by using only the two closest neighborhood relations we have not enough freedom. Otherwise using three or four neighborhood relations the situation is much more complex, and a non-symmetric distance function may appear [40,31].

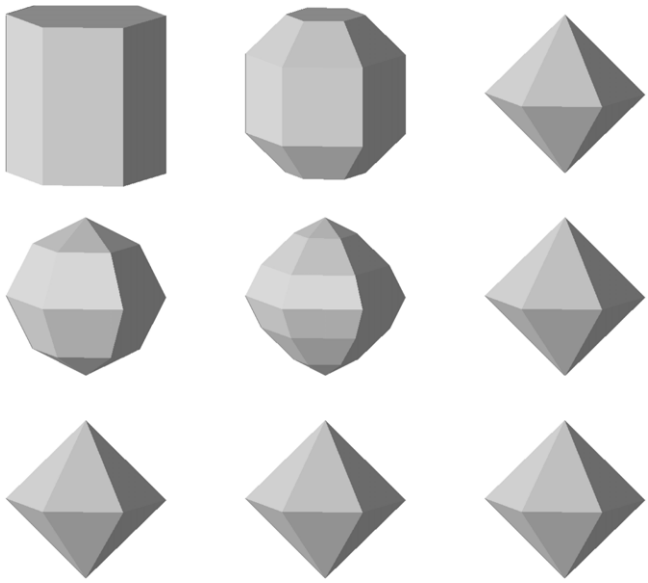

Fig. 9. Asymptotic shapes of balls in the honeycomb grid. Left to right: $\gamma = 0, 0.5, 1$. Top to bottom: $\beta/\alpha = 1, 1.5, 2$.

Fig. 10. Asymptotic shape of balls in the diamond grid for *any* parameters.

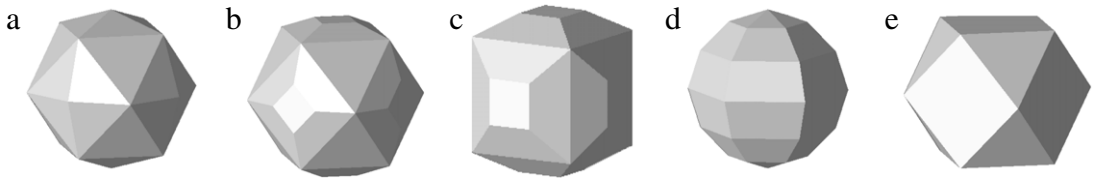

Fig. 11. Asymptotic shapes of balls in the (a) cubic, (b) fcc, (c) bcc, (d) honeycomb, and (e) diamond grids with optimal parameters (see text).

Table 2
Optimal parameters and the corresponding values of E.

Grid	β/α	γ	E
$\mathbb{Z}^3$	1.307	0	0.137
$\mathbb{F}$	1.486	0.487	0.127
$\mathbb{B}$	1.220	0.453	0.158
$\mathbb{H}$	1.313	0.424	0.148
$\mathbb{D}$	[1, 2]	[0, 1]	0.349

5. Distance transforms

The distance transform is a mapping from the image domain, a subset of the grid, to the range of the distance function. In a distance transform, each object grid point is given the distance from the closest background grid point. With the distance functions presented here, the distance values correspond to minimal-cost paths. This implies that the centers of maximal balls (the balls in an object that are not completely covered by any other ball in the object) can be computed efficiently and without errors for some digital distance functions [8,50,51] opposed to when the Euclidean distance is used [52]. The original object can be recovered from the set of centers of maximal balls by computing the *reverse* distance transform. It is straightforward to compute the reverse distance transform with digital distance functions [8]. When the Euclidean distance is used, this computation is more complex [53]. Also, when computing the constrained distance transform (the distance transform on non-convex domains, for example, the *geodesic* distance) using the Euclidean distance, a complex algorithm

based on visible points is needed [54]. The corresponding algorithm using a path-based approach is simple, fast, and easy to generalize to higher dimensions [55,56]. The algorithm presented in this section can be used to compute the constrained distance transform, but here we restrict the discussion to the unconstrained case.

The image domain is a finite subset of the grid $\mathbb{G}$ denoted $\mathbb{1}_{\mathbb{G}}$. We call the function $F : \mathbb{1}_{\mathbb{G}} \longrightarrow \mathcal{R}_d$, where $\mathcal{R}_d$ is the range of the distance function d, an *image*.

The union of the set of object grid points and the set of background grid points equals the image domain $\mathbb{1}_{\mathbb{G}}$. We denote the distance transform for path-based distances with $DT_{\mathcal{C}}$, where the subscript $\mathcal{C}$ indicates that costs are computed.

Definition 3. The *distance transform* $DT_{\mathcal{C}}$ generated by the distance function $d_{\alpha,\beta}(\cdot,\cdot;B)$ of an object $X \subset \mathbb{1}_{\mathbb{G}}$ is the mapping

$$\begin{aligned} DT_{\mathcal{C}} : \mathbb{1}_{\mathbb{G}} &\to \mathcal{R}_d \text{ defined by} \\ P &\mapsto d_{\alpha,\beta}\left(\overline{X}, P; B\right), \text{ where} \\ d_{\alpha,\beta}\left(\overline{X}, P; B\right) &= \min_{Q \in \overline{X}} \{d(Q, P)\}. \end{aligned}$$

For weighted ns-distances, the size of the neighborhood allowed in each step is determined by the *length* of the minimal cost-path (not the cost), so this value is also needed when propagating distance information. We define the auxiliary transform $DT_{\mathcal{L}}$ that holds the length of the minimal cost path at each point.

Definition 4. The transform $DT_{\mathcal{L}}$ of an object $X \subset \mathbb{G}$ is the mapping

$$\begin{aligned} DT_{\mathcal{L}} : \mathbb{1}_{\mathbb{G}} &\to \mathbb{N} \text{ defined by} \\ P &\mapsto d_{1,1}(Q, P; B), \text{ where} \\ Q \text{ is such that } d_{\alpha,\beta}(Q, P; B) &= d_{\alpha,\beta}\left(\overline{X}, P; B\right). \end{aligned}$$

Algorithm 1: Computing $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$ for weighted ns-distances by wavefront propagation.

Input: B, α, β, and an object $X \subset \mathbb{G}$.
Output: The distance transforms $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$.
Initialization: Set $DT_{\mathcal{C}}(P) \leftarrow 0$ for grid points $P \in \overline{X}$ and $DT_{\mathcal{C}}(P) \leftarrow \infty$ for grid points $P \in X$. Set $DT_{\mathcal{L}} = DT_{\mathcal{C}}$. For all grid points $P \in \overline{X}$ adjacent to X: push $(P, DT_{\mathcal{C}}(P))$ to the list L of ordered pairs sorted by increasing $DT_{\mathcal{C}}(P)$.
Notation: $\omega_{\overrightarrow{v}}$ is α if $\overrightarrow{v}$ corresponds to a 1-step and β if $\overrightarrow{v}$ corresponds to a strict 2-step.

```
while L is not empty do
    foreach P in L with smallest DT_C(P) do
        Pop (P, DT_C(P)) from L;
        foreach Q: Q, P are b(DT_L(P) + 1)-neighbors do
            if DT_C(Q) > DT_C(P) + ω_{P−Q} then
                DT_C(Q) ← DT_C(P) + ω_{P−Q};
                DT_L(Q) ← DT_L(P) + 1;
                Push (Q, DT_C(Q)) to L;
            end
        end
    end
end
```

Algorithm 1 shows how the distance transform can be computed by wavefront propagation. The algorithm is based on the Dijkstra algorithm, which has time complexity $O(n \log n)$, where n is the number of grid points in the image domain. The factor n is unavoidable, since each grid point has to be visited at least once and the factor $\log n$ is needed to keep the auxiliary data structure, the list L of ordered pairs, sorted. However, with *integer* weights, the list can be kept sorted in constant time, since only integer distance values are attained. The sorting is then similar to a *pigeonhole sort*. Also, since the wavefront contains a limited number of distance values, this approach does not lead to any significant increase in space complexity.

In [8], it is proved that Algorithm 1 gives correct results on point-lattices. The distance transform $DT_{\mathcal{C}}$ can be computed in linear time without the additional transform $DT_{\mathcal{L}}$ on point-lattices by using a look-up table [8].

5.1. Digital balls

Here we give the shape of digital balls, i.e., balls defined by the digital distance functions. Algorithm 1 was used to generate the balls in Fig. 12, where each grid point in the ball is represented by the corresponding voxel. The radii of the balls are 20α. In the implementations, we used integer approximations that give good approximations of the optimal, real-valued parameters. The main reason for using integer parameters is that it is easy and efficient to use integers in the algorithms. The parameters that were used are shown in Table 3.

Table 3
Integer approximations of the optimal parameters in Table 2 used to generate the digital balls in Fig. 12.

Grid	α	β	B	Number of voxels
$\mathbb{Z}^3$	10	13	(2)	28 533
$\mathbb{F}$	2	3	(1, 2)	33 385
$\mathbb{B}$	5	6	(1, 2)	26 015
$\mathbb{H}$	10	13	(1, 2, 1, 2, 2)	29 737
$\mathbb{D}$	1	1	(1, 2)	23 691

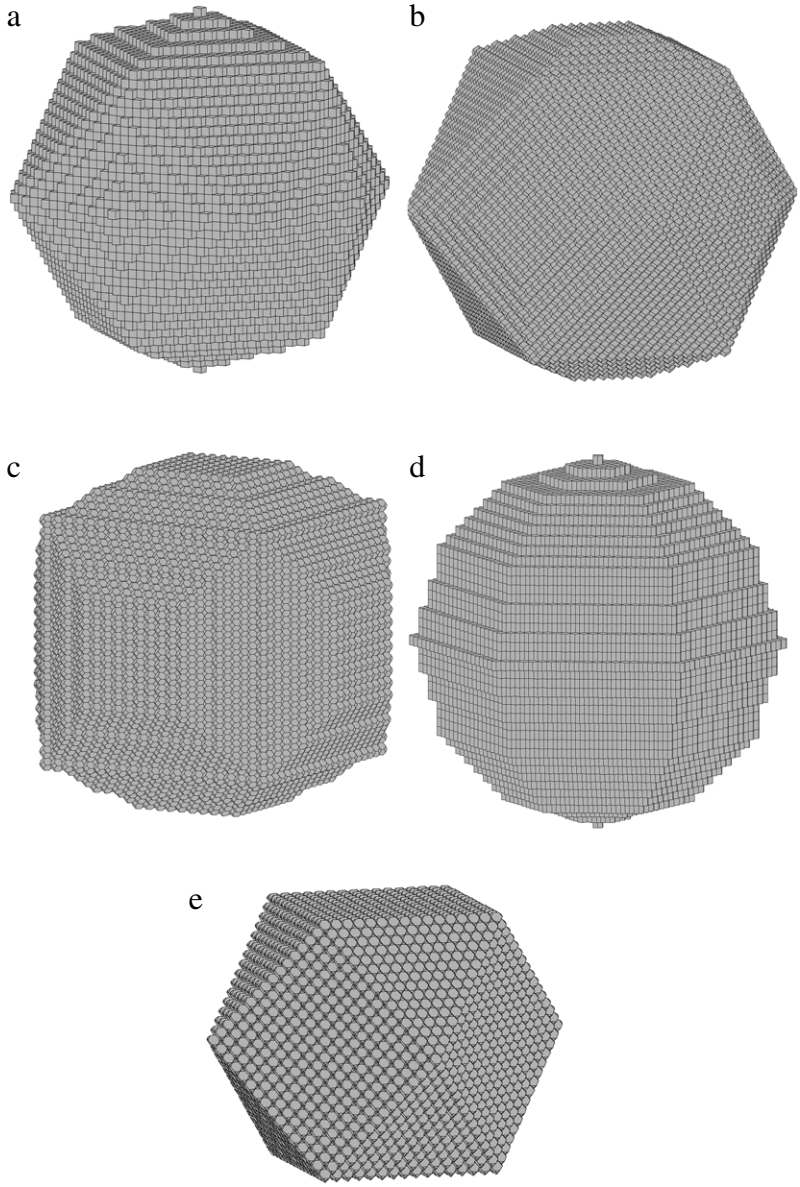

Fig. 12. Digital balls with radius 20α (=20 1-steps) in the (a) cubic, (b) fcc, (c) bcc, (d) honeycomb, and (e) diamond grids with the parameters in Table 3.

6. Conclusion and future work

With the compactness measure used here, the fcc grid has the lowest rotational dependency. The cubic grid has the second lowest value of the error function. Note that the optimal value for the cubic grid is attained for weighted distances, i.e., the neighborhood sequence does not add anything to the performance. Since the vectors corresponding to 2-neighbors

in the cubic grid are collinear with the vectors corresponding to the 2-neighbors on the fcc grid, the asymptotic shape of the digital balls are equal for these grids when the weighted distances are considered.

In our future work, we plan to apply other error functions to see if it is true that the fcc grid has lowest rotational dependency in general. The shape of the polyhedron in Fig. 11(d) seems to be a good approximation of the Euclidean sphere, so intuitively we believe that the honeycomb grid might be the best choice in some aspects. Also, in cases where the resolution is lower in the z-direction, which is often the case in CT images of living patients, the honeycomb grid handles this problem as easily as the cubic grid. The "height" of the voxels that is optimal for sampling is computed in [57] and some preliminary results on the optimal "height" for digital distance functions have been derived, but are not yet published.

By using a larger number of neighbors, the rotational dependency can be reduced. As we mentioned earlier, it is natural to include three neighborhoods for the cubic and diamond grids, since each voxel meet voxels from three neighborhoods in these grids. In [38,39], we used four neighborhood relations to define ns-distances (i.e., when $\alpha = \beta = 1$) on the diamond grid.

The algorithm for computing the (constrained) distance transform given in Section 5 gives a powerful tool for most fields of applications in image processing on these three-dimensional grids. Traditional application areas include matching [58], mathematical morphology operators [59], medial representations and skeletonization [8], etc. Recent examples of applications in which the distance transform is used are template matching [14], estimation of blood vessel width [15], and navigation in a 3D environment [16]. Digital distance functions are commonly used in these applications since the digital distance functions are efficient and the algorithms can be based on the well-known wave-propagation technique [55,56]. This is not the case for the Euclidean distance [54].

It is today a standard procedure to use three-dimensional images in, e.g., medical diagnosis. We believe that the interest in non-Cartesian, three-dimensional grids will increase, since three-dimensional images are becoming more common also in non-medical applications. Handling three-dimensional images is more efficient with grids that are optimal for representing these images.

Acknowledgements

The work of the second author is also supported by the TÁMOP 4.2.1./B-09/1/KONV-2010-0007 project. The project is implemented through the New Hungary Development Plan, co-financed by the European Social Fund and the European Regional Development Fund.

References

[1] L. Ibanez, C. Hamitouche, C. Roux, Determination of discrete sampling grids with optimal topological and spectral properties, in: Proceedings of 6th Conference on Discrete Geometry for Computer Imagery, DGCI 1996, Lyon, France, 1996, pp. 181–192.
[2] T. Meng, B. Smith, A. Entezari, A.E. Kirkpatrick, D. Weiskopf, L. Kalantari, T. Möller, On visual quality of optimal 3D sampling and reconstruction, in: GI'07: Proceedings of Graphics Interface 2007, ACM, New York, NY, USA, 2007, pp. 265–272.
[3] C. Hamitouche, L. Ibanez, C. Roux, Discrete topology of A_n^* optimal sampling grids. Interest in image processing and visualization, Journal of Mathematical Imaging and Vision 23 (3) (2005) 401–417.
[4] G.T. Herman, Geometry of Digital Spaces, Birkhäuser, Boston, 1998.
[5] B.M. Carvalho, E. Garduño, G.T. Herman, Multiseeded fuzzy segmentation on the face centered cubic grid, in: ICAPR'01: Proceedings of the Second International Conference on Advances in Pattern Recognition, Springer-Verlag, London, UK, 2001, pp. 339–348.
[6] R. Strand, G. Borgefors, Resolution pyramids on the fcc and bcc grids, in: Proceedings of 12th International Conference on Discrete Geometry for Computer Imagery, DGCI 2005, Poitiers, France, in: Lecture Notes in Computer Science, vol. 3429, Springer-Verlag, 2005, pp. 68–78.
[7] A. Entezari, Optimal sampling lattices and trivariate box splines, Ph.D. Thesis, Simon Fraser University, 2007.
[8] R. Strand, Distance functions and image processing on point-lattices: with focus on the 3D face- and body-centered cubic grids, Ph.D. Thesis, Uppsala University, Sweden, 2008. http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-9312.
[9] J.H. Conway, N.J.A. Sloane, E. Bannai, Sphere-packings, Lattices, and Groups, Springer-Verlag New York, Inc., New York, NY, USA, 1988.
[10] S. Matej, R.M. Lewitt, Efficient 3D grids for image reconstruction using spherically-symmetric volume elements, IEEE Transactions on Nuclear Science 42 (4) (1995) 1361–1370.
[11] S. Matej, G.T. Herman, A. Vardi, Binary tomography on the hexagonal grid using Gibbs priors, International Journal of Imaging Systems and Technology 9 (2–3) (1999) 126–131.
[12] R. Strand, Using the hexagonal grid for three-dimensional images: direct Fourier method reconstruction and weighted distance transform, in: Proceedings of 18th International Conference on Pattern Recognition, ICPR 2006, vol. 2, Hong Kong, 2006, pp. 1169–1172.
[13] V.E. Brimkov, R.P. Barneva, Analytical honeycomb geometry for raster and volume graphics, The Computer Journal 48 (2) (2005) 180–199.
[14] D.A. Jáuregui, P. Horain, Region-based vs. edge-based registration for 3D motion capture by real time monoscopic vision, in: Proceedings of the 4th International Conference on Computer Vision/Computer Graphics Collaboration Techniques, MIRAGE 2009, Springer-Verlag, 2009, pp. 344–355.
[15] S. Ukil, J. Reinhardt, Anatomy-guided lung lobe segmentation in X-ray CT images, IEEE Transactions on Medical Imaging 28 (2) (2009) 202–214.
[16] S. Garrido, L. Moreno, D. Blanco, Exploration of a cluttered environment using voronoi transform and fast marching, Robotics and Autonomous Systems 56 (12) (2008) 1069–1081.
[17] U. Montanari, A method for obtaining skeletons using a quasi-Euclidean distance, Journal of the ACM 15 (4) (1968) 600–624.
[18] G. Borgefors, On digital distance transforms in three dimensions, Computer Vision and Image Understanding 64 (3) (1996) 368–376.
[19] C. Fouard, R. Strand, G. Borgefors, Weighted distance transforms generalized to modules and their computation on point lattices, Pattern Recognition 40 (9) (2007) 2453–2474.
[20] S. Svensson, G. Borgefors, Digital distance transforms in 3D images using information from neighbourhoods up to $5 \times 5 \times 5$, Computer Vision and Image Understanding 88 (1) (2002) 24–53.
[21] A. Rosenfeld, J.L. Pfaltz, Distance functions on digital pictures, Pattern Recognition 1 (1968) 33–61.
[22] M. Yamashita, N. Honda, Distance functions defined by variable neighbourhood sequences, Pattern Recognition 17 (5) (1984) 509–513.
[23] P.P. Das, P.P. Chakrabarti, Distance functions in digital geometry, Information Sciences 42 (1987) 113–136.
[24] R. Strand, B. Nagy, Distances based on neighbourhood sequences in non-standard three-dimensional grids, Discrete Applied Mathematics 155 (4) (2007) 548–557.

[25] A. Rosenfeld, J.L. Pfaltz, Sequential operations in digital picture processing, Journal of the ACM 13 (4) (1966) 471–494.
[26] M. Yamashita, T. Ibaraki, Distances defined by neighbourhood sequences, Pattern Recognition 19 (3) (1986) 237–246.
[27] R. Strand, Weighted distances based on neighbourhood sequences in non-standard three-dimensional grids, in: Proceedings of 15th Scandinavian Conference on Image Analysis, SCIA 2007, Aalborg, Denmark, in: Lecture Notes in Computer Science, vol. 4522, Springer-Verlag, 2007, pp. 452–461.
[28] A. Hajdu, L. Hajdu, R. Tijdeman, General neighborhood sequences in $\mathbb{Z}^n$, Discrete Applied Mathematics 155 (2007) 2507–2522.
[29] R. Strand, B. Nagy, Weighted neighborhood sequences in non-standard three-dimensional grids—parameter optimization, in: Proceedings of 12th International Workshop on Combinatorial Image Analysis, IWCIA 2008, Buffalo, NY, in: Lecture Notes in Computer Science, vol. 4958, Springer-Verlag, 2008, pp. 51–62.
[30] B. Nagy, R. Strand, Neighborhood sequences on nd hexagonal/face-centered-cubic grids, in: Proceedings of the 13th International Workshop on Combinatorial Image Analysis, IWCIA 2009, Springer-Verlag, 2009, pp. 96–108.
[31] B. Nagy, R. Strand, Neighborhood sequences in the diamond grid—algorithms with four neighbors, in: Proceedings of the 13th International Workshop on Combinatorial Image Analysis, IWCIA 2009, Springer-Verlag, 2009, pp. 109–121.
[32] B. Nagy, Distance functions based on neighbourhood sequences, Publicationes Mathematicae Debrecen 63 (3) (2003) 483–493.
[33] B. Nagy, Distances with neighbourhood sequences in cubic and triangular grids, Pattern Recognition Letters 28 (1) (2007) 99–109.
[34] B. Nagy, Distance with generalized neighbourhood sequences in nD and ∞D, Discrete Applied Mathematics 156 (12) (2008) 2344–2351.
[35] J. Kepler, Strena seu de nive sexangula (the six-cornered snowflake) (1611).
[36] T. Hales, Cannonballs and honeycombs, Notices of the AMS 47 (4) (2000) 440–449.
[37] P. Stelldinger, R. Strand, Topology preserving digitization with FCC and BCC grids, in: Proceedings of 11th International Workshop on Combinatorial Image Analysis, IWCIA 2006, Berlin, Germany, 2006, in: Lecture Notes in Computer Science, vol. 4040, Springer-Verlag, 2006, pp. 226–240.
[38] B. Nagy, R. Strand, A connection between $\mathbb{Z}^n$ and generalized triangular grids, in: Proceedings of 4th International Symposium on Advances in Visual Computing, ISVC 2008, Part II, Springer-Verlag, 2008, pp. 1157–1166.
[39] B. Nagy, R. Strand, Non-traditional grids embedded in $\mathbb{Z}^n$, International Journal of Shape Modeling 14 (2) (2008) 209–228.
[40] B. Nagy, R. Strand, Neighborhood sequences in the diamond grid: algorithms with two and three neighbors, International Journal of Imaging Systems and Technology 19 (2) (2009) 146–157.
[41] B. Nagy, Metric and non-metric distances on $\mathbb{Z}^n$ by generalized neighbourhood sequences, in: Proceedings of 4th International Symposium on Image and Signal Processing and Analysis, ISPA 2005, Zagreb, Croatia, 2005, pp. 215–220.
[42] R. Strand, Weighted distances based on neighborhood sequences for point-lattices, Discrete Applied Mathematics 157 (4) (2009) 641–652.
[43] B. Nagy, R. Strand, Neighborhood sequences in the diamond grid, in: Image Analysis—From Theory to Applications (Proceedings of 12th International Workshop on Combinatorial Image Analysis, IWCIA 2008, Buffalo, NY), Research Publishing, Singapore, Chennai, 2008, pp. 187–195.
[44] P.P. Das, Best simple octagonal distances in digital geometry, Journal of Approximation Theory 68 (2) (1992) 155–174.
[45] P.P. Das, B.N. Chatterji, Octagonal distances for digital pictures, Information Sciences 50 (1990) 123–150.
[46] B.J.H. Verwer, Local distances for distance transformations in two and three dimensions, Pattern Recognition Letters 12 (11) (1991) 671–682.
[47] A. Hajdu, L. Hajdu, Approximating the Euclidean distance using non-periodic neighbourhood sequences, Discrete Mathematics 283 (2004) 101–111.
[48] J. Farkas, S. Baják, B. Nagy, Notes on approximating the Euclidean circle in square grids, Pure Mathematics and Applications 17 (2006) 309–322.
[49] B. Nagy, R. Strand, Approximating Euclidean distance using distances based on neighbourhood sequences in non-standard three-dimensional grids, in: Proceedings of 11th International Workshop on Combinatorial Image Analysis, IWCIA 2006, Berlin, Germany, in: Lecture Notes in Computer Science, vol. 4040, Springer-Verlag, 2006, pp. 89–100.
[50] E. Remy, E. Thiel, Medial axis for chamfer distances: computing look-up tables and neighbourhoods in 2D or 3D, Pattern Recognition Letters 23 (2002) 649–661.
[51] R. Strand, Shape representation with maximal path-points for path-based distances, in: Proceedings of 5th International Symposium on Image and Signal Processing and Analysis, ISPA 2007, Istanbul, Turkey, 2007, pp. 397–402.
[52] J. Hulin, E. Thiel, Visible vectors and discrete euclidean medial axis, Discrete and Computational Geometry 42 (4) (2009) 759–773.
[53] D. Coeurjolly, A. Montanvert, Optimal separable algorithms to compute the reverse Euclidean distance transformation and discrete medial axis in arbitrary dimension, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 437–448.
[54] D. Coeurjolly, S. Miguet, L. Tougne, 2D and 3D visibility in discrete geometry: an application to discrete geodesic paths, Pattern Recognition Letters 25 (5) (2004) 561–570.
[55] B.J.H. Verwer, P.W. Verbeek, S.T. Dekker, An efficient uniform cost algorithm applied to distance transforms, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (4) (1989) 425–429.
[56] R. Strand, F. Malmberg, S. Svensson, Minimal cost-path for path-based distances, in: Proceedings of 5th International Symposium on Image and Signal Processing and Analysis, ISPA 2007, Istanbul, Turkey, 2007, pp. 379–384.
[57] R. Strand, Interpolation and sampling on a honeycomb lattice, in: International Conference on Pattern Recognition, IEEE Computer Society, Los Alamitos, CA, USA, 2010, pp. 2222–2225.
[58] G. Borgefors, Hierarchical chamfer matching: a parametric edge matching algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (6) (1988) 849–865.
[59] P. Nacken, Chamfer metrics in mathematical morphology, Journal of Mathematical Imaging and Vision 4 (1994) 233–253.